DEVELOPMENT OF ASYNCHRONOUS SERIAL DATA COMPARATOR USING FPGA

ARWENA BINTI ABDULLAH

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor Degree of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering
Kolej Universiti Kejuruteraan & Teknologi Malaysia

APRIL, 2006

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : _____

Name : MD RIZAL BIN OTHMAN

Date : 20 APRIL 2006

Signature : _____

Author : ARWENA BINTI ABDULLAH

Date : 20 APRIL 2006

ACKNOWLEDGMENT

In the name of Allah swt the Beneficient, the Merciful…

There are just a few of the many important people who have provided enormous contribution towards the success of this thesis. I truly appreciate all their contribution and would like to express our sincere indebtedness to them . Their contribution either in the forms of comments, suggestions and even criticisms is highly appreciated. For me, this work is not my achievement rather it comprises of effort and cooperation from them.

Firstly, I would like to extend our sincere appreciation to our academic associates, especially to the supervisor Encik Md Rizal B Othman for his continuous support and encouragement. Secondly, to the Dean of the Faculty of Electrical & Electronic Engineering. Besides that, I would like to thanks our lecturers , staff of the laboratory of electrical and electronic engineering, my parents and to my friends for their valuable support, contribution and help.

Finally, thanks go to the anonymous editor and manuscript reviewer from books and internet who had help to improve the final version of this thesis.

To all those mentioned and many more, I personally indebted to your invaluable cooperation. I take full responsibility for any errors, omissions, and negligence in this thesis. My sincere hope that, after reading this thesis it would give positive feedback about this project.

ABSTRAK

Penggunaan FPGA yang lebih mudah dan semakin meluas dalam aplikasi harian membuka peluang kepada para pelajar untuk lebih mempelajari penggunaan dan aplikasi FPGA dalam bidang elektronik. Terdapat banyak aplikasi yang boleh diimplementasikan ke dalam FPGA dan salah satu contohnya adalah dalam bidang Penghantaran perbandingan data sesiri secara asynchronous

Secara ringkasnya, projek ini bertujuan untuk mengkonfigurasi dan mengawal papan FPGA sebagai litar luaran tambahan untuk berkomunikasi antara MAX233 dengan PC untuk penghantaran data secara sesiri untuk menyimpan dan mengeluarkan data dalam pengunaan perbandingan ID kod dengan ID nombor.Program ini dibina dalam proses digit system dimana data adalah dalam bit bit.Data akan dikenali dengan 8 bit dan akan diperkenalkan sebagai data bit yang akan menentukan ID nombor dengan kod yang dihantar.Setiap bit ID nombor akan disimpan di dalam flipflop yang berbeza oleh 8 daftar anjak.Sekiranya keluaran flipflop adalah sama dengan ID nombor,litar akan menghasilkan paparan logik tinggi.

ABSTRACT

The wide and simpler usage of FPGA in everyday applications opened the door for students to study its usage and applications in electronics. There are many applications that can be implemented in FPGA and one of them is in developing asynchronous serial data comparator using FPGA.

This project is involved the design, implementation and test of a digital logic system by the application of the FPGA as the programmable gate logics where will be implemented as the program to communicate with the MAX233 and the PC for serial data comparator transferring and to control in storing and accessing the data in the application of comparison ID code and ID number. The program developed is processed in digital form that is the data bit generated is in digital. It will recognizes the 8 bits and it represent as the data bits that determine the ID number and compare the code. Each bit of the ID number will stored in the different flip-flop of 8 shift register. If each flip-flop output matches the ID number, the circuitry produces a high output. This circuitry then will burn in to the ALTERA and will communicate with the data that transmit.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In today's world the term digital has become part of our everyday vocabulary because of the dramatic way that digital circuit and digital techniques have become so widely used in almost all areas of life: computers, automation ,robots, medical science and technology, transportation, entertainment, space exploration, and on and on. One of the most common operation that occur in any digital system is the transmission of data from one place to another .The data can be transmitted over a distance as small as a fraction of an inch on the same circuit board, or over distance of many miles. The data that is transmitted is in binary form and generally represented as voltage at the outputs of a sending circuit that are connected to the inputs of receiving circuit. There are two basic methods for digital data transmission: parallel and serial. By far the most common use of flip-flops is for the storage of data or information. These data are generally stored in group of FFs called registers. The transfer operations are performed in synchronous or asynchronous transfer Synchronous serial transmission requires that the sender and receiver share a clock with one another, or that the sender provide a strobe or other timing signal so that the receiver knows when to "read" the next bit of the data. Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver.

In this project, the application of the FPGA is use as the programmable gate logics where will be implemented as the program to communicate with the MAX233 and the PC for serial data comparator transferring.

This project involved the design, compile, implementation and simulation of a system using the MAX+pluss II. In this project application, it will working as the serial data comparator to compare the ID code and ID number. This project processed in digital form that is the data bit generated is in digital. It will recognizes the 8 bits and it represent as the data bits that determine the ID number and compare the code .Each bit of the ID number will stored in the different flip-flop of 8 shift register. If each flip-flop output matches the ID number, the circuitry produces a high output. This will produces the LED and the buzzer on. at the FPGA board.

The transmitted serial bit data in baud rate 96000act as the input to FPGA. Baud is a measurement of transmission speed in asynchronous communication .So this project can used many applications such as in security entry system. This system allows us to deactivate the alarm so that we can enter the building. If the correct digit has been entered on the keypad, the 8 bits on the inputs of the comparator are same as the 8 bits on the keypad and the comparator produces a HIGH on the output.

1.2 OBJECTIVES

The objective of this project is to develop a asynchronous serial data comparator using field programmable that implemented on FPGA. The FPGA, acts as an external hardware is used as a medium for communication between the PC with DB9 and the external MAX233 that will be connected directly with the FPGA. The logic circuit will be created by following the stages in the ***MAX+plusII*** software The other software will be developed is in Visual Basic for the serial port communication between the FPGA and PC. To accomplish such objective, both of the software must work well and communicate with each other in order for the transfer of reading and writing of data file to be successful.

Other than the objectives above, this project is to show how the FPGA works and how it can be a great tool for a microcontroller board in integrated circuit applications. Understanding its connections and the downloading process also will be learnt from this project. After all the downloading in the ALTERA chip and testing are being done, its intended function to communicate with the PC.

1.3 SCOPE

The project is expected to determine the ID code number using asynchronous data and will recognizes the 8 bits and it represent as the data bits that determine the ID number and compare the code using Max+pluss. Each bit of the ID number will store in the different flip-flop of 8 shift register. If each flip-flop output matches the ID number, the circuitry produces a high output to the circuitry.

1.4 PROBLEM STATEMENT

The idea of problem is based on comparison in operation in the system of compare ID code for example in ID number invalid or valid. By using FPGA, the ID number will be recognizes the ID code and compare it in the circuitry then produces a high output .This problem make to develop of asynchronous data comparator using FPGA to transmit the data. At this stage, objective and project scope clearly clarify to solve the problem.

CHAPTER II

LITERATURE REVIEW

The literature review is the second step to overview and study about the program flow by using the block diagram. The study done briefly to understand the implementation on FPGA using a programmable logic device to develop and to answer what, why, and how about the operation. The study used to next step of specification and design according from the source and the related project.

2.1 FPGA

FPGA stands for **F**ield-**P**rogrammable **G**ate **A**rray. It is a technology for making integrated circuits. The key difference from other chip technologies is that it is *field programmable*, meaning that you get to make the chip do what you want without having to get an external manufacturer. This means much less risk because if you make a mistake you do not have to wait for weeks to get another chip made, and you do not have to spend a lot of money to make the fix. All you do is modify your design and re-program your chip, much like you fix a C program, recompile it, load it into memory, and run it.

This capability provides lots of opportunities for doing interesting things, including just building chips full of logic. It makes a lot of the previous difficulties of building hardware easier. Some even call this *soft* hardware.

For example, you can now think of building compute engines that do their computations by configuring hardware solutions rather than executing a bunch of instructions.

(Stephen D. Brown, 1992)

2.2 PROGRAM FLOW

The first phase is development of the system block diagram of asynchronous serial data comparator using FPGA. The figure shown below is giving a draft idea on the flow of system. Further and detail explanations will be given in this section.

The data send from the PC is in the serial data with 8 bit data binary number. The connection using the DB9 as the serial communication part with the FPGA board through MAX 233 as the medium to transmit the data using the 96000 baud rate.

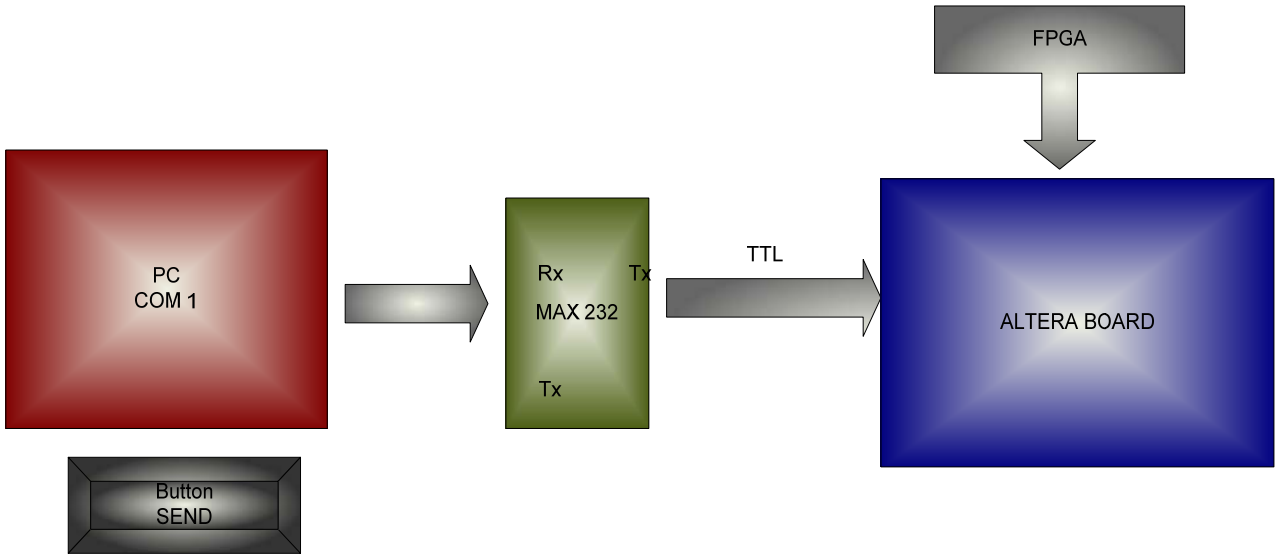

Figure 2.1: Block Diagram of Asynchronous Serial Data Comparator

FPGA

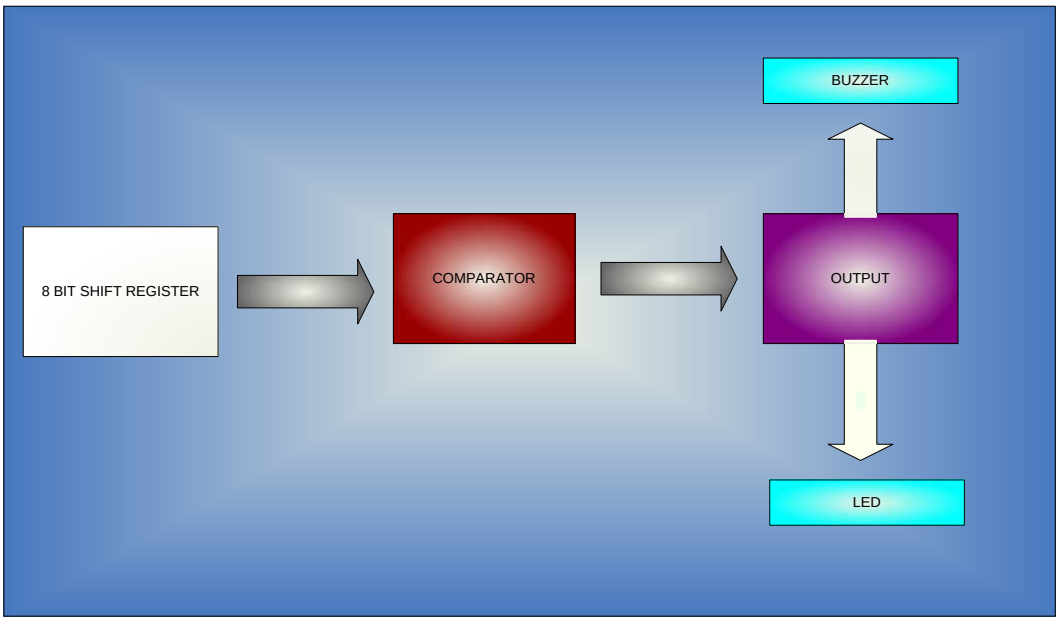

Figure 2.2: Block Diagram of Design Flow

2.2.1 Shift Register:

A shift register is a storage circuit where data is input serially from one end or in parallel and emerges from the other end after a specified number of clock cycles. In digital logic circuit, a one-bit shift register can be built using J-K flip-flop or D flip-flop. “D” is the input data, clock is the enable signal of J-K or D flip-flop. After one clock cycle, the value of D is shifted out as output Q. Other signals can be used to enhance basic operation of the shift register. They are “reset” and “preset” signal and “p” as presetted value. Using this 8-bit shift register is a convenient way to map a serial stream of signals to parallel output

2.2.2 Comparing Circuitry:

The purpose of this circuitry is to determine the ID code that has been transmitted and to compare this code to the ID number . Each bit of the ID number is stored in a different flip-flop of an eight-bit shift register in order to decode and determine the ID number. The output of each flip-flop will be compared to the ID number for each flipflop. If each flip-flop output matches the preset number, the circuitry produces a high output toup the tone generating circuitry. If the ID codes do not match, the output of the decoding logic remains low.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION OF FPGA

With the introduction of the **Field Programmable Gate Array** (‘FPGA’ - a configurable- logic chip) in the early 80’s, the hardware engineer was empowered to implement chip-level designs in silicon without having to fabricate a chip. As these devices and their software tools matured, the use of FPGAs expanded from testing and verifying digital designs to in-system use. This overview describes the fundamentals as well as current uses of this technology.

FPGAs perform the function of a custom LSI circuit, like a gate array, and are user programmable. The most significant advantage of using FPGA devices is the ability to produce a prototype logic design, implementing it in silicon within hours, while conventional gate array devices can take months and many dollars to develop and produce working silicon. Since their introduction, FPGAs have continued to increase in useable gate count, while decreasing in price. They are currently being used as glue logic, for test / verification logic in system designs, for adaptable system designs and more recently as co processing devices. FPGAs are also used to emulate other component architectures, and are applicable for rapid prototyping. With the next generation of SRAM based FPGAs (designed with computing in mind) a whole new generation of computing applications will result.

A field-programmable gate array or FPGA is a semiconductor device containing programmable logic components and programmable interconnects. The programmable logic components can be programmed to duplicate the functionality of basic logic gates (such as AND, OR, XOR, NOT) or more complex combinatorial functions such as decoders or simple math functions. In most FPGAs, these programmable logic components (or logic blocks, in FPGA parlance) also include memory elements, which may be simple flip-flops or more complete blocks of memories.

A hierarchy of programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer, somewhat like a one-chip programmable breadboard. These logic blocks and interconnects can be programmed after the manufacturing process by the customer/designer (hence the term "field-programmable") so that the FPGA can perform whatever logical function is needed.

FPGAs are generally slower than their application-specific integrated circuit (ASIC) counterparts, can't handle as complex a design, and draw more power. However, they have several advantages such as a shorter time to market, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. Vendors may offer less flexible versions of their FPGAs that are cheaper. The development of these designs is made on regular FPGAs and then migrated into a fixed version that more resembles an ASIC due to lack of ability to modify the design once it is committed. Another alternative is complex programmable logic devices CPLD.

3.2 APPLICATIONS

Applications of FPGAs include DSP, software-defined radio, aerospace and defense systems, ASIC prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, and a growing range of other areas. FPGAs originally began as competitors to CPLDs and competed in a similar space, that of glue logic for PCBs. As their size, capabilities and speed increased they began to take over larger and larger functions to the state where they are now marketed as competitors for full systems on chips. They now find applications in any area or algorithm that can make use of the massive parallelism offered by their architecture.

3.3 OVERVIEW FPGA CONSTRUCTION

There are four main categories of FPGAs currently commercially available: symmetrical array, row-based, hierarchical PLD, and sea-of-gates as shown in Figure 3.1.

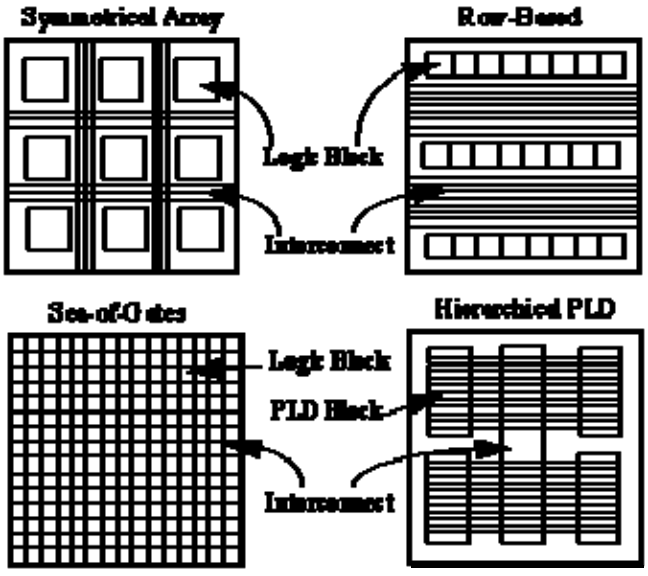

Figure 3.1: Classes of FPGA

In all of these FPGAs the interconnections and how they are programmed vary. Currently there are four technologies in use. They are: static RAM cells, anti-fuse, EPROM transistors, and EEPROM transistors.

3.3.1 Static RAM Technology -- In the Static RAM FPGA programmable connections are made using pass=transistors, transmission gates, or multiplexers that are controlled by SRAM cells. The advantage of this technology is that it allows fast in-circuit reconfiguration. The major disadvantage is the size of the chip required by the RAM technology.

3.3.2 Anti-Fuse Technology -- An anti-fuse resides in a high-impedance state; and can be programmed into low impedance or "fused" state. A less expensive than the RAM technology, this device is a program once device.

3.3.3 EPROM / EEPROM Technology -- This method is the same as used in the EPROM memories. One advantage of this technology is that it can be reprogrammed without external storage of configuration; though the EPROM transistors cannot be re-programmed in-circuit. The following table shows some of the characteristics of the above programming technologies.

3.4 THE FPGA

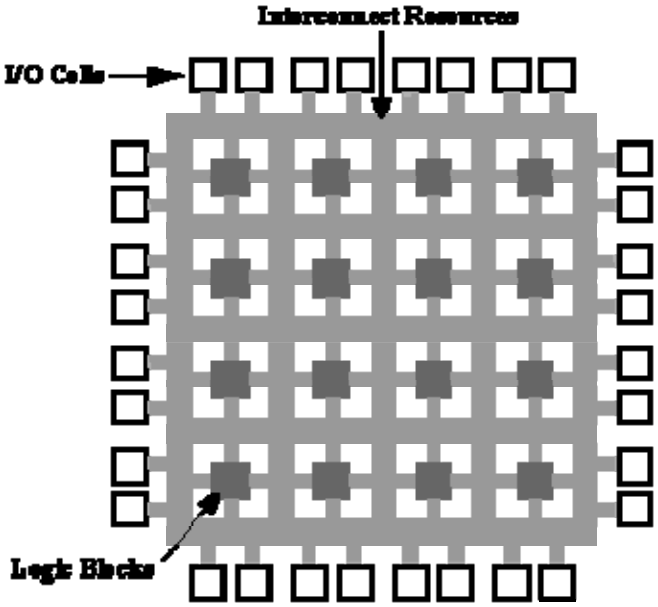

Figure 3.2: The FPGA structure

The FPGA has three major configurable elements: configurable logic blocks (CLBs), input/output blocks, and interconnects. The CLBs provide the functional elements for constructing user's logic (Figure 3.2). The IOBs provide the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs onto the appropriate networks. Customized configuration is established by programming internal static memory cells that determine the logic functions and internal connections implemented in the FPGA.

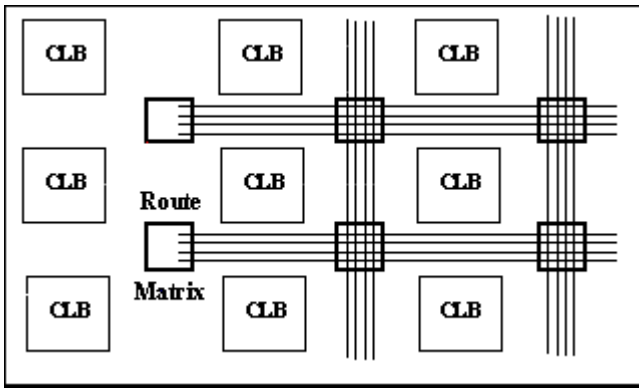

Figure 3.3: CLBs interconnect

Figure 3.3 depicts a FPGA with a two-dimensional array of logic blocks that can be interconnected by interconnect wires. All internal connections are composed of metal segments with programmable switching points to implement the desired routing. An abundance of different routing resources is provided to acheive efficient automated routing. There are four main types of interconnect, three are distinguished by the relative length of their segments: single-length lines, double-length lines and longlines. (NOTE: The number of routing channels shown in the figure are for illustration purposes only; the actual number of routing channels varies with the array size.) In addition, eight global buffers drive fast, low-skew nets most often used for clocks or global control signals.

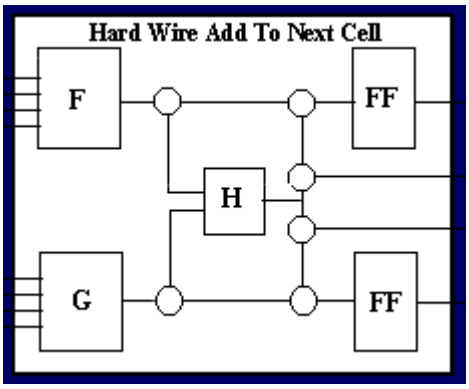

Figure 3.4: Configurable Logic Blocks